Blatt 14 - Inhalte

1. CORS
2. JavaScript Utility
3. JavaScript Webanwendung 1

Same-Origin Policy und CORS

- Schutzfunktion, JavaScript darf nur auf Inhalte vom gleichen Host zugreifen
  - Beispiel: JavaScript kommt von localhost/index.html
    - localhost/foo/bar/something => OK
    - google.de/ => ERROR
    - localhost:1337/remote/ => ERROR
- Intention: Schutz vor dem Nachladen beliebiger Daten (z.B. andere Website unter gefälschter URL anzeigen, Schadcode nachladen, …)
- Problem: Was ist mit legitimen Anfragen?
  - Wetterdaten
  - Fahrplandaten
  - Tarifdaten
  - …

CORS

- Cross-Origin Resource Sharing
- Ziel: Kontrolliertes „Umgehen“ der Same-Origin Policy
- Server bestimmt, unter welchen Umständen fremde JavaScript Anfragen gültig sind
- Zwei Möglichkeiten:
  - Simple Requests (nur GET/HEAD/POST, nur bestimmte Header, Content-Types, …)
  - Complex Requests (alles andere)

CORS Preflight

- Bei nicht simplen CORS Requests überprüft der Browser zuerst, ob der Request zulässig ist
- Dies geschieht über einen HTTP Options Request

```
OPTIONS / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:51.0)
Gecko/20100101 Firefox/51.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Access-Control-Request-Method: GET
Access-Control-Request-Headers: content-type
Origin: null
DNT: 1
Connection: keep-alive
```